データベース処理のデータフロー言語 (情報の記憶と利用に関する理論的研究)

AUTHOR(S):

田中, 譲

CITATION:

田中, 譲. データベース処理のデータフロー言語 (情報の記憶と利用に関する理論的研究). 数理解析研究所講究録 1981, 423: 106-115

ISSUE DATE:

1981-04

URL:

http://hdl.handle.net/2433/102579

数理解析研究所講究録
423巻 1981年 106-115

106

データベース処理のデータフロー言語

北海道大学・工学部
田中 譲

1. 序論

現在著者等が開発中のデータストリーム処理方式データベースマシンのためのデータフロー言語について試案を述べる。このマシンはサーチエンジン（SEE：SEarch Engine）とソートエンジン（SOE：SOrt Engine）と名付けた2種の機能モジュールを用いて，データベース処理のボトルネックとなるデータの転送と，サーチ，ソートのデータベースにおける基本演算の処理とを重畳させ，データベース処理の高速化を図ろうというものである。[1][2][3] 本論文では，このマシンのアーキテクチュア，SEEとSOEのアーキテクチュアに関する説明は省略する。

我々は，このマシンのシステム言語を3つのレベルに分けて考えている。最上位レベルは，更新演算を含むように拡張した関係代数言語で，第2レベルは，これよりも少々記述のレベルが低く，よく似ている原始関係代数言語，そして最下位レベルが，データベースマシンの各構成モジュールの機能とよく対応のとれているデータストリーム言語である。

CODASYL DMLインターフェースやQBEインターフェース等は，システム言語の最上位にある関係代数言語との論理的なマッピングの問題なので，ここでは触れないでおく。

関係代数言語とデータストリーム言語の間に原始関係代数言語を置いたのは，オブジェクトコードの最適化や，スケジューリングの問題をできる限り，関係代数言語と原始関係代数言語との間のマッピングの問題として論理的に把えたいという心積もりからである。本論文では，最適化およびスケジューリングの問題を避け，原始関係代数言語とデータストリーム言語の間の変換について述べる。さらに，更新演算についても本論文では触れないでおく。

2. 原始関係代数言語

原始関係代数は関係代数と同じ演算を持つが，属性を属性名で参照することができず，かわりに指標を用いる。
R(A, B, C)，S(D, E, F)，T(G, H, I)に対して，関係代数式の例とこれに対応する原始関係代数式をいくつか列挙する。

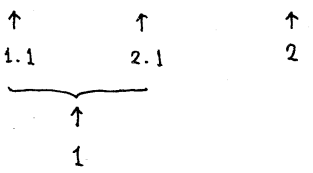
R[A, B]	[1, 2]R
R[B=F]S	[2=3]RS
R[C='u']	{3='u'}R

$R[A=B]$ $\quad\quad$ $\{1=2\}R$

$(R[B=F]S)[C,D]$ $\quad\quad$ $[3.1, 1.2][2=3]RS$

注, Rを1, Sを2で表わし, A, B, Cを1.1, 2.1, 3.1で表わす.

$((R[B=F]S)[E=I]T)[C,D,H]$ $\quad\quad$ $[3.1.1, 1.2.1, 2.2][2.2=3]$ $[2=3]RST$



指標 $i_0.i_1.\cdots.i_n$ に対して, i_0 をヘッド, $i_1.\cdots.i_n$ をテイルと呼ぶ。指標 α のヘッドとテイルを $h(\alpha)$, $t(\alpha)$ で表わす.

通常の結合演算 $R[X=Y]S$ の他に, 半結合演算

$$R\langle X=Y]S \triangleq \{r \mid r\in R \text{ s.t. } \exists s\in S \quad r.X=s.Y\}$$

が考慮する.

原始関係代数言語では, プロジェクションをとらない限り, 関係中のタプルの各値を見ることはできないとする。

3. データストリーム言語

データタイプ α のデータストリームとは, データタイプ α に属するデータの有限系列のことである。タイプ α のデータ

$a_0, a_1, \cdots, a_n$ がこの順に並んだデータストリームを，$\langle a_0, a_1, \cdots, a_n \rangle$ で表わす．データタイプ α のデータストリームの集合を α^* で表わす．

リストを以下のように定義する．

(1) a_i をタイプ α_i のデータとするとき，$(a_0, a_1, \cdots, a_n)$ はタイプ $(\alpha_0, \alpha_1, \cdots, \alpha_n)$ のリストである．

(2) b_i をタイプ β_i のデータないしはリストとするとき，$(b_0, b_1, \cdots, b_n)$ はタイプ $(\beta_0, \beta_1, \cdots, \beta_n)$ のリストである．

(3) (1)(2) を有限回適用して得られるものだけがリストである．

タイプ α のリストのストリームの集合を α^{**} で表わす．データタイプ α の要素のみを用いて得られるリストのストリームのすべてを $\amalg_\alpha$ で表わすことにする．i を正整数とするとき，

$$\cdot i\,(a_1, a_2, \cdots, a_n) = \begin{cases} a_i & i \leq n \\ \text{undefined} & \text{otherwise.} \end{cases}$$

と定義する．したがって，例えば

$$\cdot 2\cdot 3\,(a, (b, c), (e, (f, g), h)) = (f, g)$$

である。

データストリーム言語で扱うデータタイプには，関係名のタイプ$\mathbb{R}$，データベースに現われる値のタイプ$\mathbb{D}$，非負整数のタイプ$\mathbb{N}$と，ストリーム型$\mathbb{D}^*$，$\mathbb{N}^*$，$\mathbb{L}_{\mathbb{D}}$がある。

以下にデータストリーム言語の関数演算子を列挙する。

$!: \mathbb{R} \to \mathbb{N}^*$

関係Rのタプル数をnとするとき，

$!R = <0, 1, 2, \cdots, (n-1)>$

である。

$?: \mathbb{R} \to (\mathbb{N} \to (\mathbb{N}^* \to \mathbb{D}^*))$

$?R\,i\,<i_1, i_2, \cdots, i_j>$

iがRの属性数より大きいとき空ストリーム，

そうでなければRのj番目のタプルのi属性の値をa_jとするとき

$<a_{i_1}, a_{i_2}, \cdots, a_{i_j}>$

とする。

$\uparrow: \mathbb{D}^* \to \mathbb{N}^* \quad (\downarrow: \mathbb{D}^* \to \mathbb{N}^*)$

$\uparrow <a_0, a_1, \cdots, a_n> \quad (\downarrow <a_0, a_1, \cdots, a_n>)$

$a_0, a_1, \cdots, a_n$を昇順（降順）にソートしたものを$a_{i_1}, a_{i_2}, \cdots, a_{i_{n+1}}$とするとき，

$< i_1, i_2, \cdots, i_{n+1} >$

を値とする。

$S_\theta : \mathbb{D} \to (\mathbb{D}^* \to \mathbb{N}^*)$

$S_\theta\, v\, < a_0, a_1, \cdots, a_n >$

$a_i \theta v$ となる i を順に $i_0 < i_1 < \cdots < i_k$ とするとき

$< i_0, i_1, \cdots, i_k >$

を値とする。

$R_\theta : \mathbb{D}^* \to (\mathbb{D}^* \to \mathbb{N}^*)$

$R_\theta < a_0, a_1, \cdots, a_n > < b_0, b_1, \cdots, b_m >$

$a_i \theta b_i$ となる $0 \le i \le \min(n, m)$ を $i_0 < i_1 < \cdots < i_k$ とするとき、

$< i_0, i_1, \cdots, i_k >$

を値とする。

$X^L_\theta : \mathbb{D}^* \to (\mathbb{D}^* \to (\mathbb{N}, \mathbb{N})^*)$

$X^R_\theta : \mathbb{D}^* \to (\mathbb{D}^* \to (\mathbb{N}, \mathbb{N})^*)$

$X^L_\theta < a_0, a_1, \cdots, a_n > < b_0, b_1, \cdots, b_m >$

$(X^R_\theta < a_0, a_1, \cdots, a_n > < b_0, b_1, \cdots, b_m >)$

$i = 0, 1, 2, \cdots, m$ の順に、$a_{j_i} \theta b_i$ となる最小（最大）の j_i を求め (i, j_i) を出力する。$< a_0, a_1, \cdots, a_n >$ は $a_0 \le a_1 \le \cdots \le a_n$ の場合に限る。

$A : \mathbb{L}_D \to (\mathbb{N}^* \to \mathbb{L}_D)$

$A\langle a_0, a_1, \cdots, a_n\rangle\langle i_0, i_1, \cdots, i_m\rangle$

$j = 0, 1, 2, \cdots, m$ の順に，$i_j \leq n$ であれば a_{i_j} を出力する。

$\epsilon : \mathbb{N}^* \to (\mathbb{N}^* \to (\mathbb{N}^* \to (\mathbb{N}, \mathbb{N})^*))$

被演算項である3つのタイプ $\mathbb{N}$ のデータストリームの各々の非 k 要素を i_1, i_2, i_3 とする。これら非 k 要素が $i_2 \leq i_3$ のとき，これらに対して，サブストリーム

$\langle (i_1, i_2), (i_1, i_2+1), \cdots, (i_1, i_3)\rangle$

を出力する。

$\mathbb{1} : \mathbb{L}_D \to \mathbb{L}_D$

$s \in \mathbb{L}_D$ に対し，$\mathbb{1}s = s$ である。

$\cdot : \mathbb{N} \to (\mathbb{L}_D \to \mathbb{L}_D)$

$i \geq 1$ に対し，$\cdot i$ は前に定義したものに等しい。

$P : \mathbb{L}_D \to (\mathbb{L}_D \to \mathbb{L}_D)$

$P\langle a_0, a_1, \cdots, a_n\rangle\langle b_0, b_1, \cdots, b_m\rangle$

$= \langle (a_0, b_0), (a_1, b_1), \cdots, (a_{\min(n,m)}, b_{\min(n,m)})\rangle$

さらに K，C を，

$K = \lambda xyz\ Pxz\ yz$, $C = \lambda xyz.\ xz\ yz$

と定義する．これらを用いた式がデータストリーム言語のプログラムである．

ここに述べたデータストリーム言語の各演算子は，著者等が開発中のデータベースマシンの各モジュールによって直接実行される．例えば↑，↓はSOEで処理され，?，X^L_σ，X^R_σ，Aは SEEで処理される．[3] これらの事柄の詳細は省略する．

4. 原始関係代数言語とデータストリーム言語の変換

原始関係代数言語のある式において，指標σの指す属性を持つ関係を$r(\sigma)$で表わすことにする．原始関係代数言語の各演算子は，データストリーム言語の演算子を用いて以下のように展開できる．x，yはタイプ $\mathbb{L}_{\mathbb{N}}$ の変数とする．

関係 R	! R
射影 $[\sigma_1, \sigma_2, \cdots, \sigma_n]$	$\lambda x.\ (?r(\sigma_1)\sigma_1 x\ ,\ ?r(\sigma_2)\sigma_2 x, \cdots, ?r(\sigma_n)\sigma_n x)$
選択 $\{\sigma\ \theta\ v\}$	$\lambda x.\ A\,x\,S_\theta\,v\,?r(\sigma)\sigma x$
制約 $\{\sigma\ \theta\ \sigma'\}$	$\lambda x.\ A\,x\,R_\theta\,C\,(?r(\sigma)\sigma)(?r(\sigma')\sigma')\,x$
半結合 $<\sigma\ \theta\ \sigma']$	$\lambda xy.A x .1\ X^L_\theta A?r(\sigma')\sigma' y \uparrow ?r(\sigma')\sigma' y\ ?r(\sigma)\sigma x$
結合 $[\sigma\ \theta\ \sigma']$	$\lambda xy.\ K(Ax.1)(Ay.2)\,C(\uparrow ?r(\sigma)\sigma x.1)(.1)$

$X_\theta A ?r(\sigma)\circ x \uparrow ?r(\sigma)\circ x \;\; ?r(\sigma')\circ' y$

$X_\theta \triangleq \lambda x y.\ \in(.1\, X_\theta^L x y)(.2\, X_\theta^L x y)(.2\, X_\theta^R x y)$

上述の関係を用いることにより，原始関係代数言語の任意の式をデータストリーム言語に変換することが可能となる．このときのオブジェクトプログラムは，中間結果を関係として求めることなく最終結果を与える．

上の定義において，例えば「$\circ\theta\circ'$」のオブジェクトコードで，母式のxとyを入れ換えたものが「$\circ\theta\circ'$」のオブジェクトコードと考えることができ，しかし一般に両者は処理に要する時間が異なる．ここでオブジェクトコードの最適化，スケジューリングの問題が生じる．しかし，今の例で後者のオブジェクトコードを必要とするのであれば「$\circ\theta\circ'$」xyとする所を「$\circ'\theta\circ$」yxとすればよい．このことは，他の演算子についても言え，最適化とスケジューリングの問題は，データストリーム言語のレベルより一段上に引き上げられ，原始関係代数言語のレベルで議論することができる．したがって，上述のように，各演算子のオブジェクトコードを固定してしまってよい．

5. 結論

本論文では、現在著者等が開発中のデータストリーム処理方式データベースマシンのデータフロー言語として、原始関係代数言語と、マシンの構成要素の機能に対応した演算子からなるデータストリーム言語の2つを示し、前者の演算子の後者の演算子を用いた一意な展開形を与えた。これにより、検索処理に限っては、関係代数とデータストリーム言語の美しい対応関係がとれた。

今後の課題としては、更新処理を含めた拡張がある。さらに、データストリーム言語の体系そのものの単純化が課題の1つとしてある。このためには、組み合わせ論理で用いられるコンビネーターを用いた体系化等が考えられる。

参考文献

(1) Y. Tanaka, Y. Nozaka, and A. Masuyama, "Pipeline Searching and Sorting Modules as Components of a Data Flow Database Computer," Proc. IFIP Congress 80, 1980 (Tokyo), pp 427-432.

(2) 田中譲, "データストリーム処理方式のデータベースコンピュータ," 情報処理研資, 記号処理 12-14, 1980 pp 97-103.

(3) 田中譲他, "データストリーム処理方式のデータベースコンピュータ," 情報処理全国大会, 1980, pp 493-494.